

Entendendo o Paradigma Orientado por Objetos

- Maneira mais natural de decompor os problemas.
- Agrega nas Abstrações as Funcionalidades e as Características.
- Naturalmente já **class**ificamos as coisas.

Entendendo o Paradigma Orientado por Objetos



Entendendo o Paradigma Orientado por Objetos

Mamíferos

Eqüinos



Caninos



Felinos

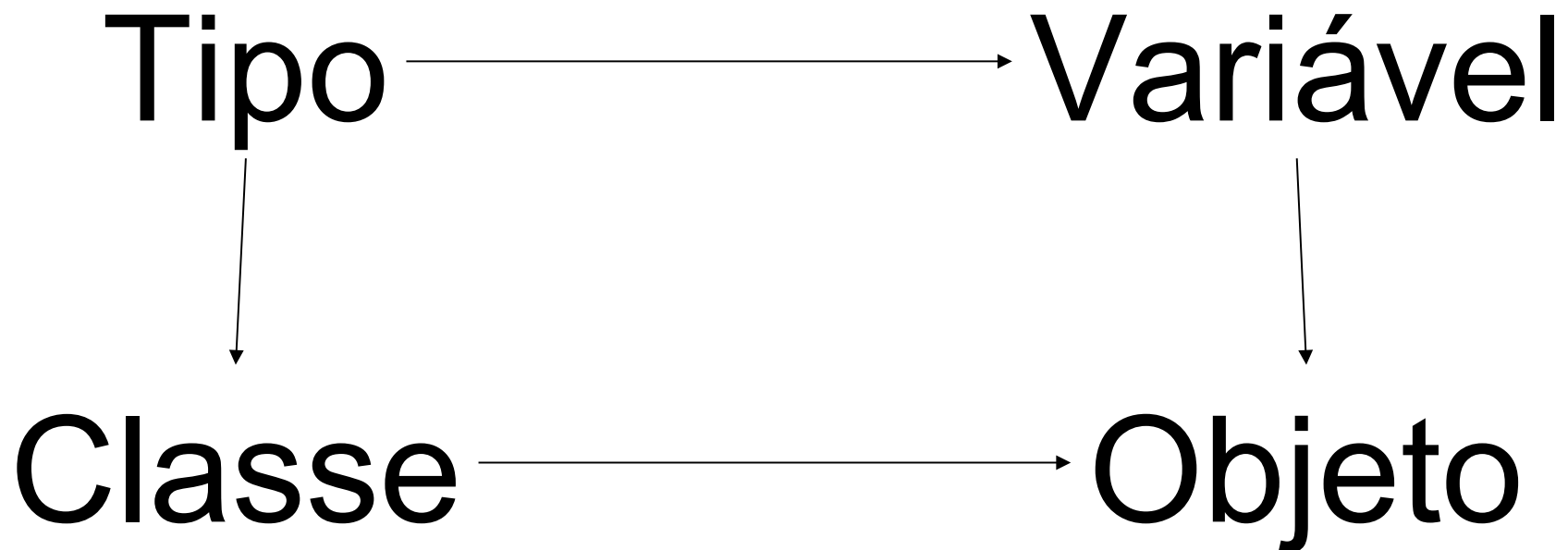


Entendendo o Paradigma Orientado por Objetos

- Mamíferos pode ser considerado a CLASSE.
- Os diferentes animais são considerados sub-classes de mamíferos.
- Cada um com suas funcionalidades e características.
- Seu cãozinho de estimação pode ser considerado uma instância da sub-classe caninos.

Entendendo a Classe

- Forma ou Molde para criar objetos!
- A classe está para o objeto assim como o tipo está para a variável.



Criando um classe de mamíferos

```
class Mamiferos
```

```
{
```

```
    boolean sangueQuente = true;
```

```
    void movimento()
```

```
{
```

```
        System.out.print("Procurando Comida...");
```

```
}
```

Atributo da
Classe

Método

Tipo de
Retorno

```
public static void main(String[] args)
```

```
{
```

```
    Mamiferos exp;
```

```
    exp = new Mamiferos();
```

```
    exp.movimento();
```

```
    System.out.println("Sangue Quente " +  
        exp.sangueQuente);
```

```
}
```

```
}
```

Referência
para objeto

Instância
um Novo
Objeto

`movimento ()` Função ou Método?

- Em POO a funcionalidade associada a uma determinada classe é chamada de Método.
- Um método pode retornar um tipo de dado simples ou mesmo um objeto.(referência)
- Métodos podem receber parâmetros de entrada: `movimento(int x, int y, int z)`



Exercitando...



- Crie uma classe chamada Operação que:
 - Tenha um método chamado soma
 - Tenha um método chamado subtração
 - Cada método deve retornar seu resultado correspondente através do comando **return**(e).
 - A impressão dos resultados deve ficar por conta do método **main**



Exercitando...



- Crie um arquivo chamado VariasClasses.java que contenha:
 - Uma declaração de uma classe que representa um carro.
 - Ele terá rodas, vidros, cor, motor
 - Ele pode acelerar; frear; ligar. Pode se identificar como carro.
 - Uma classe que represente um motorista.
 - Ele terá nome, CNH
 - Compile a classe que você gerou. Discuta o resultado obtido.

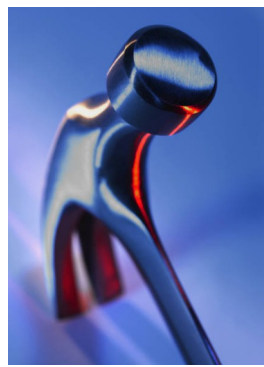


Exercitando...



- Crie(ou atualize) o arquivo chamado Mamiferos.java e faça:
 - Tenha declaração de mamífero do exemplo anterior. Assim como o método void main.
 - Acrescente a este arquivo mais duas novas referências (rato e gato) a objetos do tipo mamíferos
 - Compare se as referências rato e gato são iguais (==). São? Tente explicar.
 - Faça com que a referencia do exemplo, exp, seja igual a referência do rato. Explique!
- Explique o que aconteceu...

O Método Construtor



- Método especial chamado sempre que um objeto é instanciado. (Criado)
- Normalmente utilizado para *SETAR* parâmetro iniciais dos objetos.
- **SEMPRE** possui o mesmo nome da classe!

O Método Construtor de Mamíferos

```
class Mamiferos
{
    boolean sangueQuente, vive;
    Mamiferos()
    {
        sangueQuente = true;
        vive = true;
        System.out.println("Vivendo!");
    }
    void movimento()
    {
        System.out.print("Procurando Comida...");
    }

    public static void main(String[] a)
    {
        Mamiferos exp = new Mamiferos();
    }
}
```



Exercitando...



- Defina uma classe Pessoa, cujos objetos representam uma pessoa, contendo as seguintes informações: nome, sexo, RG, e data de nascimento. Não esqueça do construtor. Crie ainda um método toString(), que exiba tais informações.
- Crie construtores para as classes Carro e Motorista.

E/S em Janelas

```
import javax.swing.*;

class JanelasDeDialogo1
{
    public static void main (String[] a)
    {
        String entrada =
        JOptionPane.showInputDialog("Entre com um
        valor inteiro: ");
        int x = Integer.parseInt(entrada);
        JOptionPane.showMessageDialog(null, "Valor
        + 20 = " + (x+20));
    }
}
```



Exercitando...



- Modifique a classe Operação e a classe Pessoa, para que ambas utilizem recursos de E/S em janelas.
- Modifique o quadrado de arrobas, para que sua saída seja uma janela de mensagem.

Herança

- Mecanismo que permite a criação de sub-classes.
- Facilita o aproveitamento das classes já criadas.
- A classe de mais alto nível hierárquico de Java é **Object**.

Testando a Herança...

...

```
class Caninos extends Mamiferos
{
    void latir()
    {
        System.out.print("Latindo...");
    }
    void cavar()
    {
        System.out.print("Cavando...");
    }
    public static void main(String[] a)
    {
        Caninos rex = new Caninos();
        rex.latir();
        rex.movimento(); //??
        rex.latir();

    }
}
```

Exercitando...



- Acrescente um método construtor para a classe caninos. Sugestão: passe uma String representando o nome do novo canino que está sendo instanciado.
- Defina uma classe Aluno, subclasse de Pessoa que contenha: número de matrícula, curso, data de ingresso na universidade. Redefina o seu construtor.